DevDoodle.net

A developer network

October 15, 2015

Brian Blair

DevDoodle

Aims

- Share code, get feedback
- Learn new skills
- Ask and answer questions
- Create lasting repository of knowledge

Aims

- Share code, get feedback
- Learn new skills
- Ask and answer questions

- Create lasting repository of knowledge

DevDoodle

Aims

- Share code, get feedback
- Learn new skills
- Ask and answer questions

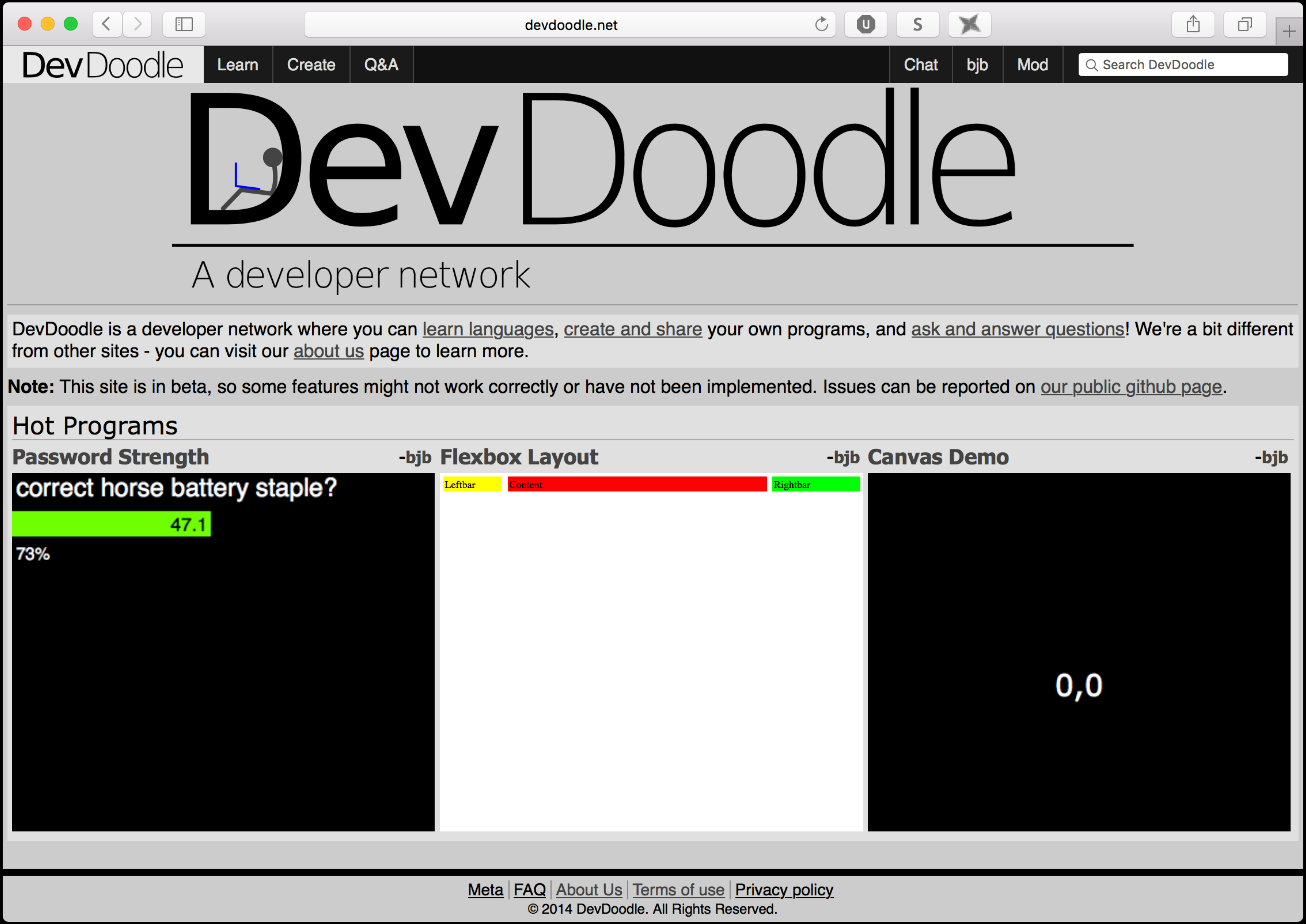
$\left.\begin{matrix}\\ \\ \\ \end{matrix}\right\}$ Immediate benefits

---

- Create lasting repository of knowledge

$\longrightarrow$ Long-term help for search engine users

DevDoodle



DevDoodle

A developer network

DevDoodle is a developer network where you can learn languages, create and share your own programs, and ask and answer questions! We're a bit different from other sites - you can visit our about us page to learn more.

Note: This site is in beta, so some features might not work correctly or have not been implemented. Issues can be reported on our public github page.

Hot Programs

Password Strength -bjb

correct horse battery staple?

47.1

73%

Flexbox Layout -bjb

Leftbar Content Rightbar

Canvas Demo -bjb

0,0

Meta | FAQ | About Us | Terms of use | Privacy policy

© 2014 DevDoodle. All Rights Reserved.

DevDoodle

Competitors

- Khan Academy
- Corsera
- Stack Overflow
- Mozilla Developer Network
- Codecademy

DevDoodle

Competitors' Weaknesses

- Incomplete product
- Clash of user groups
- Shallow tutorials, lack of advanced material
- S:N

DevDoodle

Target Users

- Younger programmers
- Some previous experience
- Able to debug own code
- eg, advanced high school CS students

DevDoodle

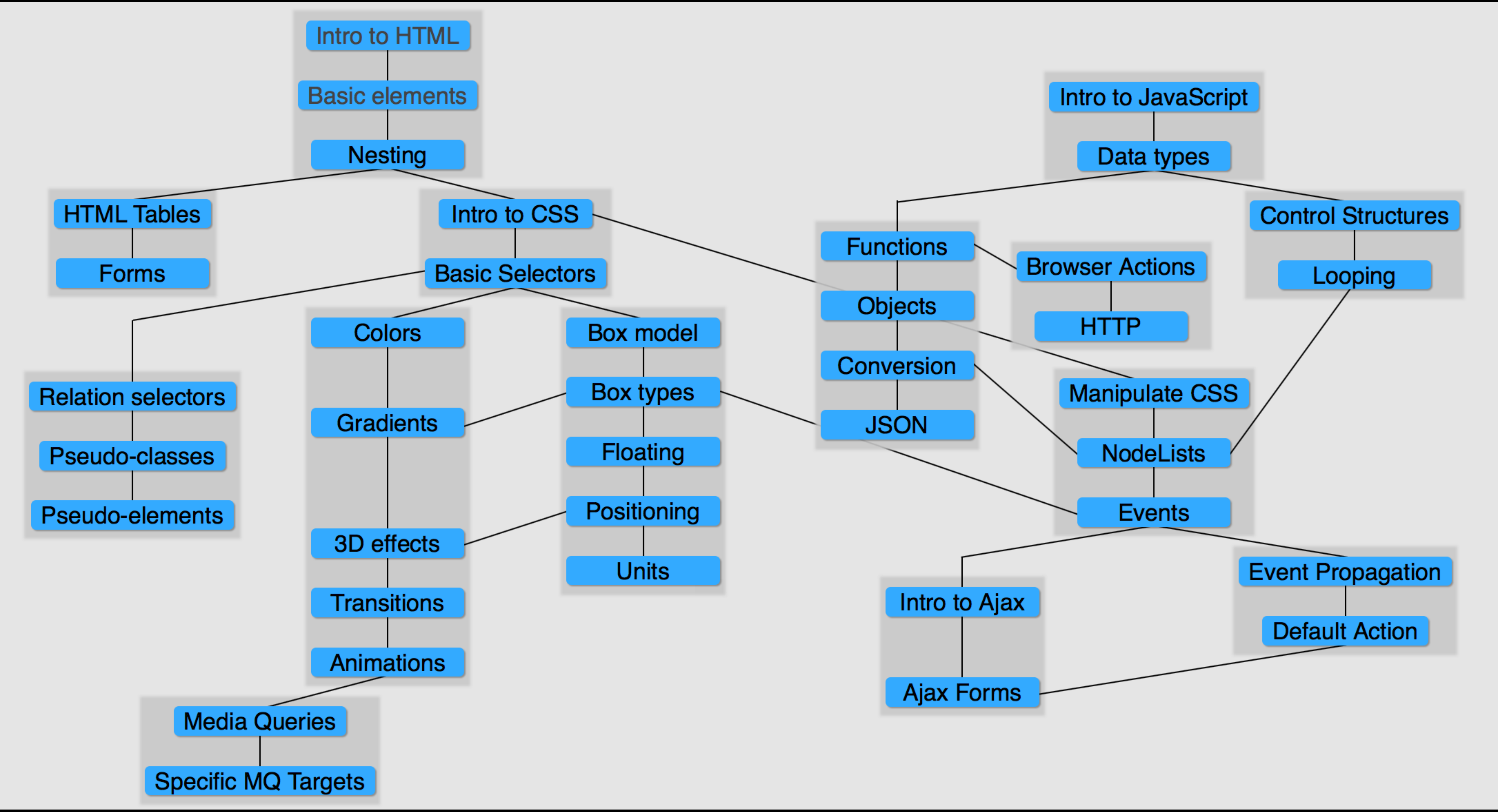
Sections

- Learn
- Create
- Q&A
- Chat

DevDoodle

Sections

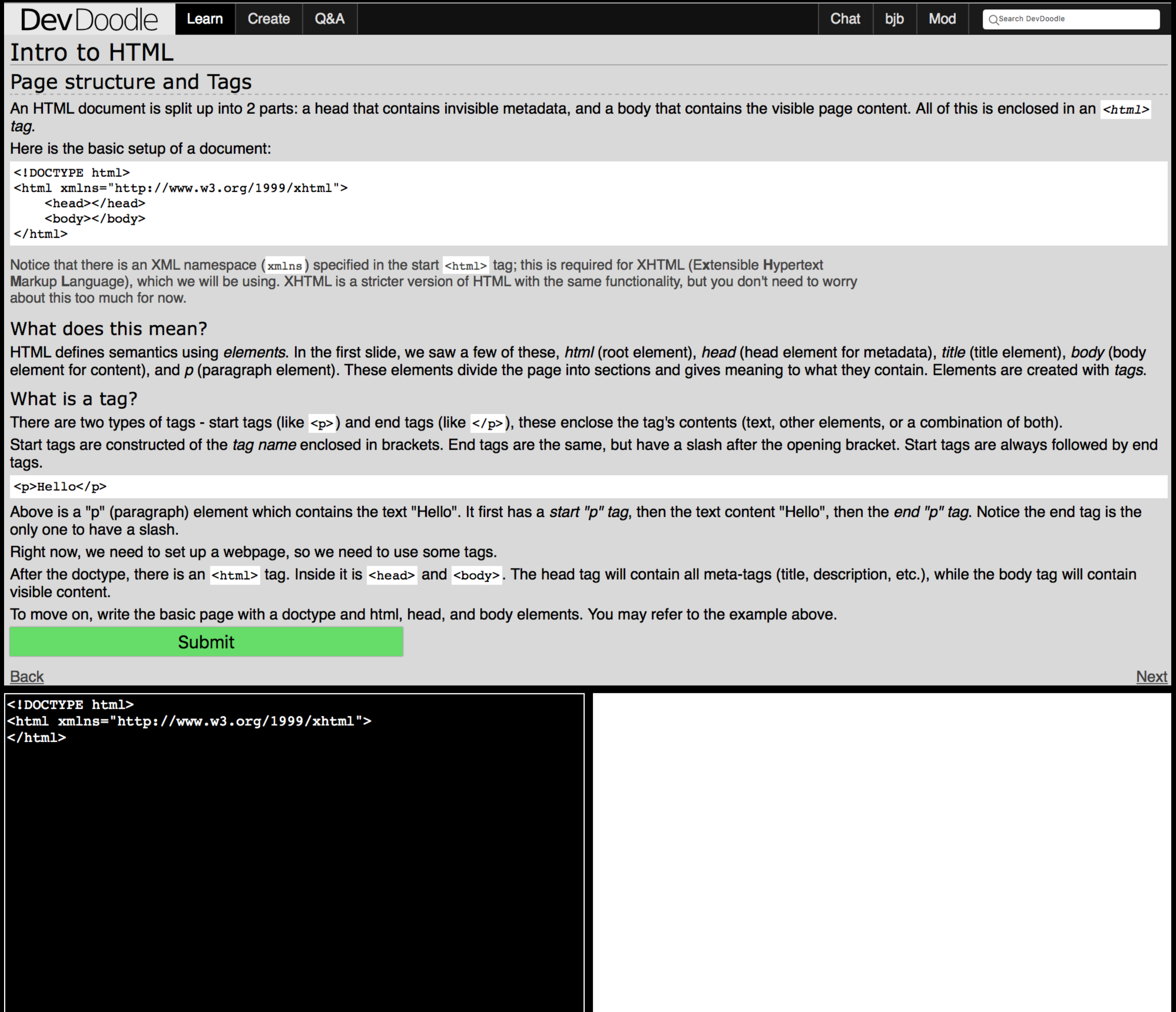
- Learn



DevDoodle

Sections

- Learn



Intro to HTML

Page structure and Tags

An HTML document is split up into 2 parts: a head that contains invisible metadata, and a body that contains the visible page content. All of this is enclosed in an *`<html>` tag*.

Here is the basic setup of a document:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
    <head></head>
    <body></body>
</html>
```

Notice that there is an XML namespace (`xmlns`) specified in the start `<html>` tag; this is required for XHTML (**Ex**tensible **H**ypertext **M**arkup **L**anguage), which we will be using. XHTML is a stricter version of HTML with the same functionality, but you don't need to worry about this too much for now.

What does this mean?

HTML defines semantics using *elements*. In the first slide, we saw a few of these, *html* (root element), *head* (head element for metadata), *title* (title element), *body* (body element for content), and *p* (paragraph element). These elements divide the page into sections and gives meaning to what they contain. Elements are created with *tags*.

What is a tag?

There are two types of tags - start tags (like `<p>`) and end tags (like `</p>`), these enclose the tag's contents (text, other elements, or a combination of both).

Start tags are constructed of the *tag name* enclosed in brackets. End tags are the same, but have a slash after the opening bracket. Start tags are always followed by end tags.

```
<p>Hello</p>
```

Above is a "p" (paragraph) element which contains the text "Hello". It first has a *start "p" tag*, then the text content "Hello", then the *end "p" tag*. Notice the end tag is the only one to have a slash.

Right now, we need to set up a webpage, so we need to use some tags.

After the doctype, there is an `<html>` tag. Inside it is `<head>` and `<body>`. The head tag will contain all meta-tags (title, description, etc.), while the body tag will contain visible content.

To move on, write the basic page with a doctype and html, head, and body elements. You may refer to the example above.

Submit

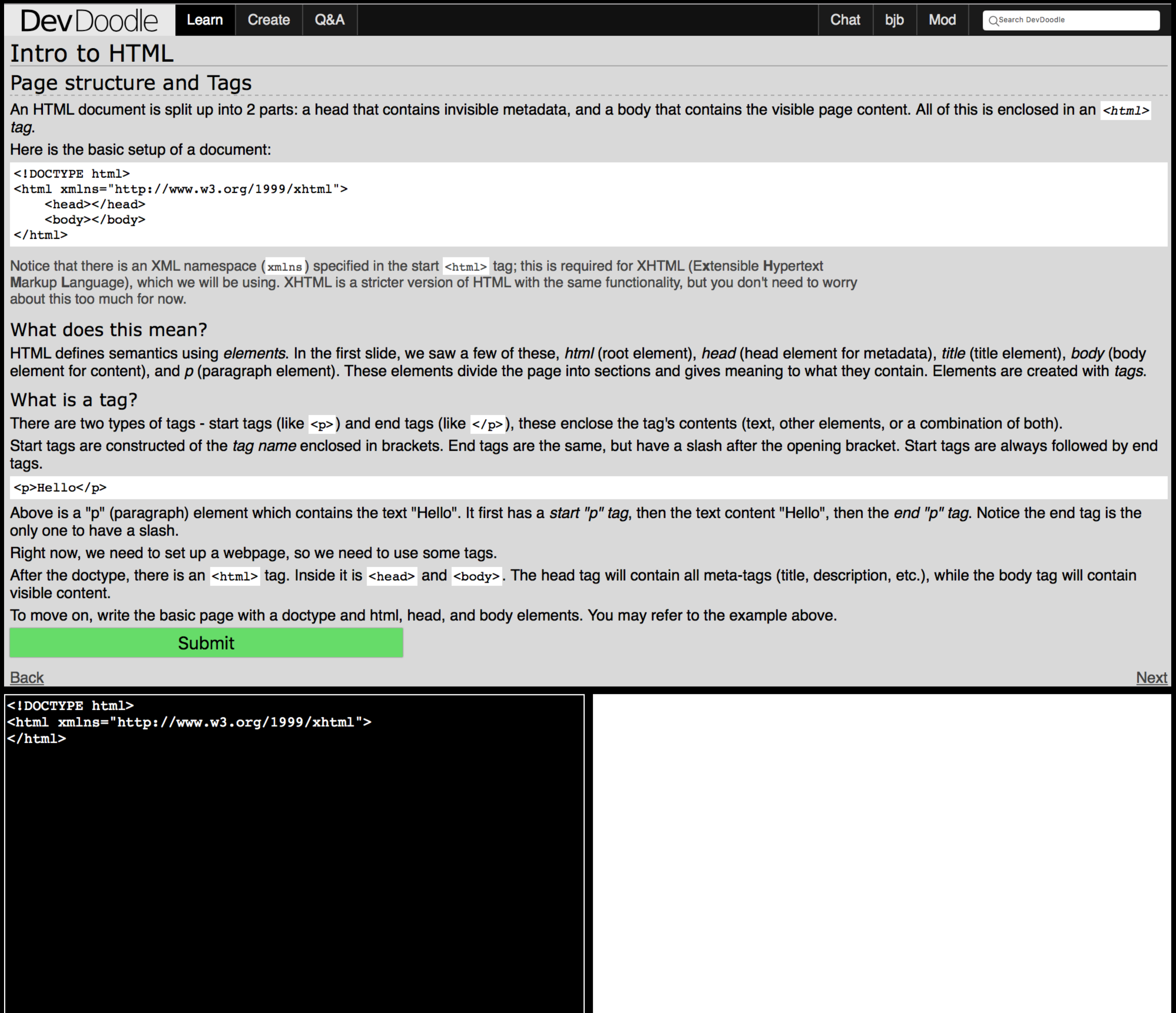
Back Next

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
</html>
```

DevDoodle

Sections

- Learn



Intro to HTML

Page structure and Tags

An HTML document is split up into 2 parts: a head that contains invisible metadata, and a body that contains the visible page content. All of this is enclosed in an `<html>` *tag*.

Here is the basic setup of a document:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
    <head></head>
    <body></body>
</html>
```

Notice that there is an XML namespace (`xmlns`) specified in the start `<html>` tag; this is required for XHTML (**Ex**tensible **H**ypertext **M**arkup **L**anguage), which we will be using. XHTML is a stricter version of HTML with the same functionality, but you don't need to worry about this too much for now.

What does this mean?

HTML defines semantics using *elements*. In the first slide, we saw a few of these, *html* (root element), *head* (head element for metadata), *title* (title element), *body* (body element for content), and *p* (paragraph element). These elements divide the page into sections and gives meaning to what they contain. Elements are created with *tags*.

What is a tag?

There are two types of tags - start tags (like `<p>`) and end tags (like `</p>`), these enclose the tag's contents (text, other elements, or a combination of both).

Start tags are constructed of the *tag name* enclosed in brackets. End tags are the same, but have a slash after the opening bracket. Start tags are always followed by end tags.

```
<p>Hello</p>
```

Above is a "p" (paragraph) element which contains the text "Hello". It first has a *start "p" tag*, then the text content "Hello", then the *end "p" tag*. Notice the end tag is the only one to have a slash.

Right now, we need to set up a webpage, so we need to use some tags.

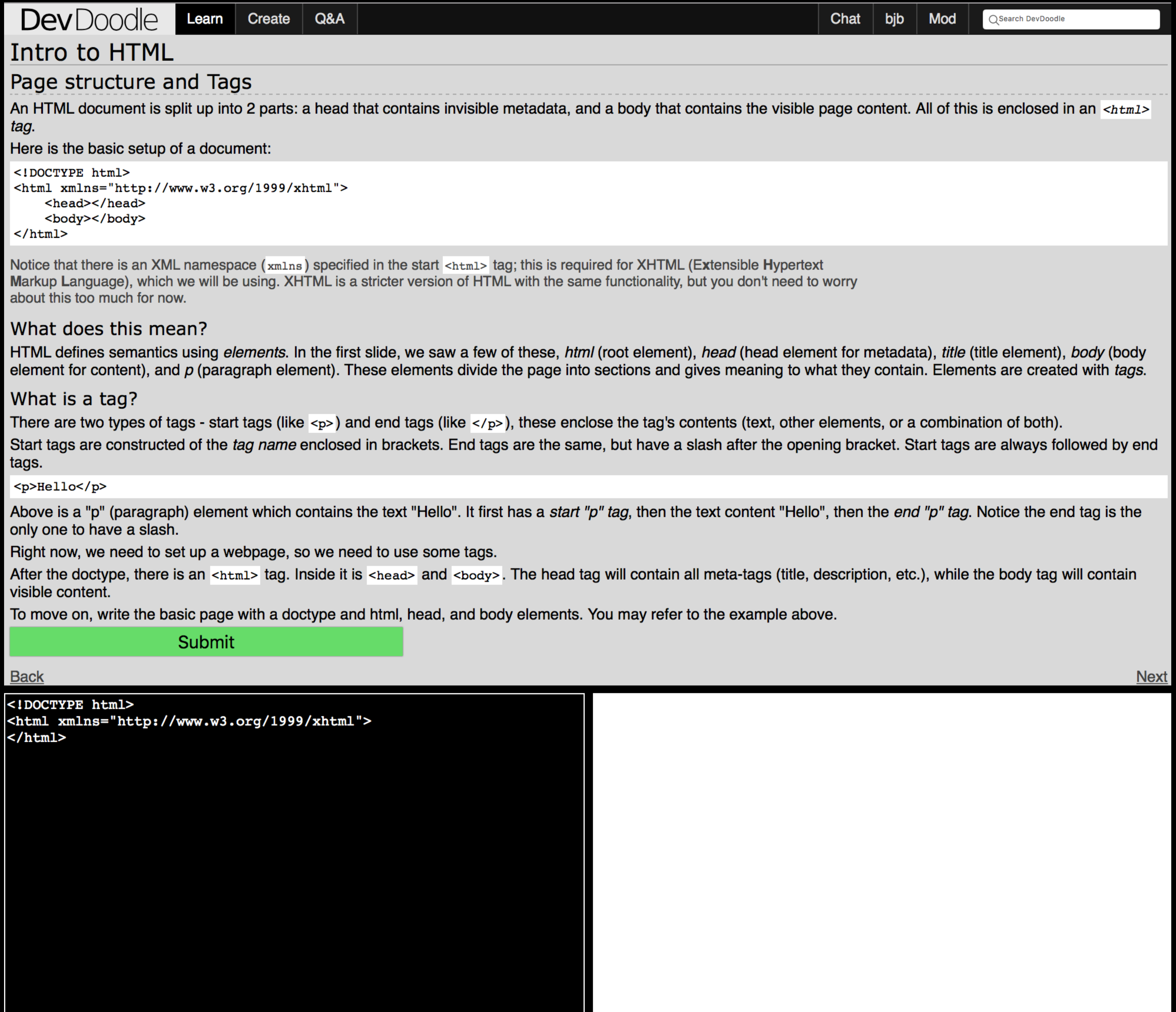
After the doctype, there is an `<html>` tag. Inside it is `<head>` and `<body>`. The head tag will contain all meta-tags (title, description, etc.), while the body tag will contain visible content.

To move on, write the basic page with a doctype and html, head, and body elements. You may refer to the example above.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
</html>
```

Sections

- Learn



Intro to HTML

Page structure and Tags

An HTML document is split up into 2 parts: a head that contains invisible metadata, and a body that contains the visible page content. All of this is enclosed in an `<html>` *tag*.

Here is the basic setup of a document:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
    <head></head>
    <body></body>
</html>
```

Notice that there is an XML namespace (`xmlns`) specified in the start `<html>` tag; this is required for XHTML (**Ex**tensible **H**ypertext **M**arkup **L**anguage), which we will be using. XHTML is a stricter version of HTML with the same functionality, but you don't need to worry about this too much for now.

What does this mean?

HTML defines semantics using *elements*. In the first slide, we saw a few of these, *html* (root element), *head* (head element for metadata), *title* (title element), *body* (body element for content), and *p* (paragraph element). These elements divide the page into sections and gives meaning to what they contain. Elements are created with *tags*.

What is a tag?

There are two types of tags - start tags (like `<p>`) and end tags (like `</p>`), these enclose the tag's contents (text, other elements, or a combination of both).

Start tags are constructed of the *tag name* enclosed in brackets. End tags are the same, but have a slash after the opening bracket. Start tags are always followed by end tags.

```
<p>Hello</p>
```

Above is a "p" (paragraph) element which contains the text "Hello". It first has a *start "p" tag*, then the text content "Hello", then the *end "p" tag*. Notice the end tag is the only one to have a slash.

Right now, we need to set up a webpage, so we need to use some tags.

After the doctype, there is an `<html>` tag. Inside it is `<head>` and `<body>`. The head tag will contain all meta-tags (title, description, etc.), while the body tag will contain visible content.

To move on, write the basic page with a doctype and html, head, and body elements. You may refer to the example above.

Submit

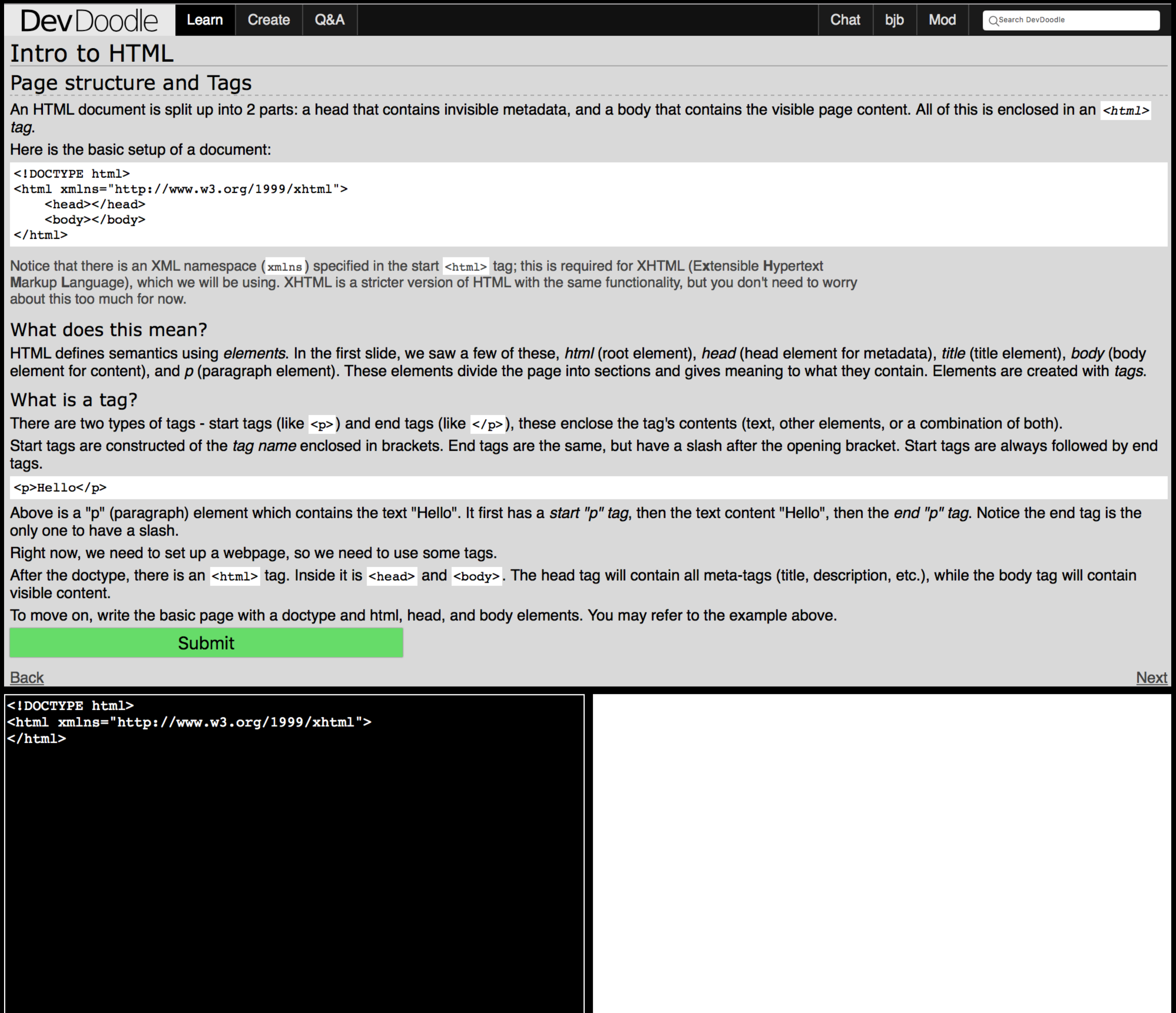
Back Next

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
</html>
```

DevDoodle

Sections

- Learn



Intro to HTML

Page structure and Tags

An HTML document is split up into 2 parts: a head that contains invisible metadata, and a body that contains the visible page content. All of this is enclosed in an `<html>` *tag*.

Here is the basic setup of a document:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
    <head></head>
    <body></body>
</html>
```

Notice that there is an XML namespace (`xmlns`) specified in the start `<html>` tag; this is required for XHTML (**Ex**tensible **H**ypertext **M**arkup **L**anguage), which we will be using. XHTML is a stricter version of HTML with the same functionality, but you don't need to worry about this too much for now.

What does this mean?

HTML defines semantics using *elements*. In the first slide, we saw a few of these, *html* (root element), *head* (head element for metadata), *title* (title element), *body* (body element for content), and *p* (paragraph element). These elements divide the page into sections and gives meaning to what they contain. Elements are created with *tags*.

What is a tag?

There are two types of tags - start tags (like `<p>`) and end tags (like `</p>`), these enclose the tag's contents (text, other elements, or a combination of both).

Start tags are constructed of the *tag name* enclosed in brackets. End tags are the same, but have a slash after the opening bracket. Start tags are always followed by end tags.

```
<p>Hello</p>
```

Above is a "p" (paragraph) element which contains the text "Hello". It first has a *start "p" tag*, then the text content "Hello", then the *end "p" tag*. Notice the end tag is the only one to have a slash.

Right now, we need to set up a webpage, so we need to use some tags.

After the doctype, there is an `<html>` tag. Inside it is `<head>` and `<body>`. The head tag will contain all meta-tags (title, description, etc.), while the body tag will contain visible content.

To move on, write the basic page with a doctype and html, head, and body elements. You may refer to the example above.

Submit

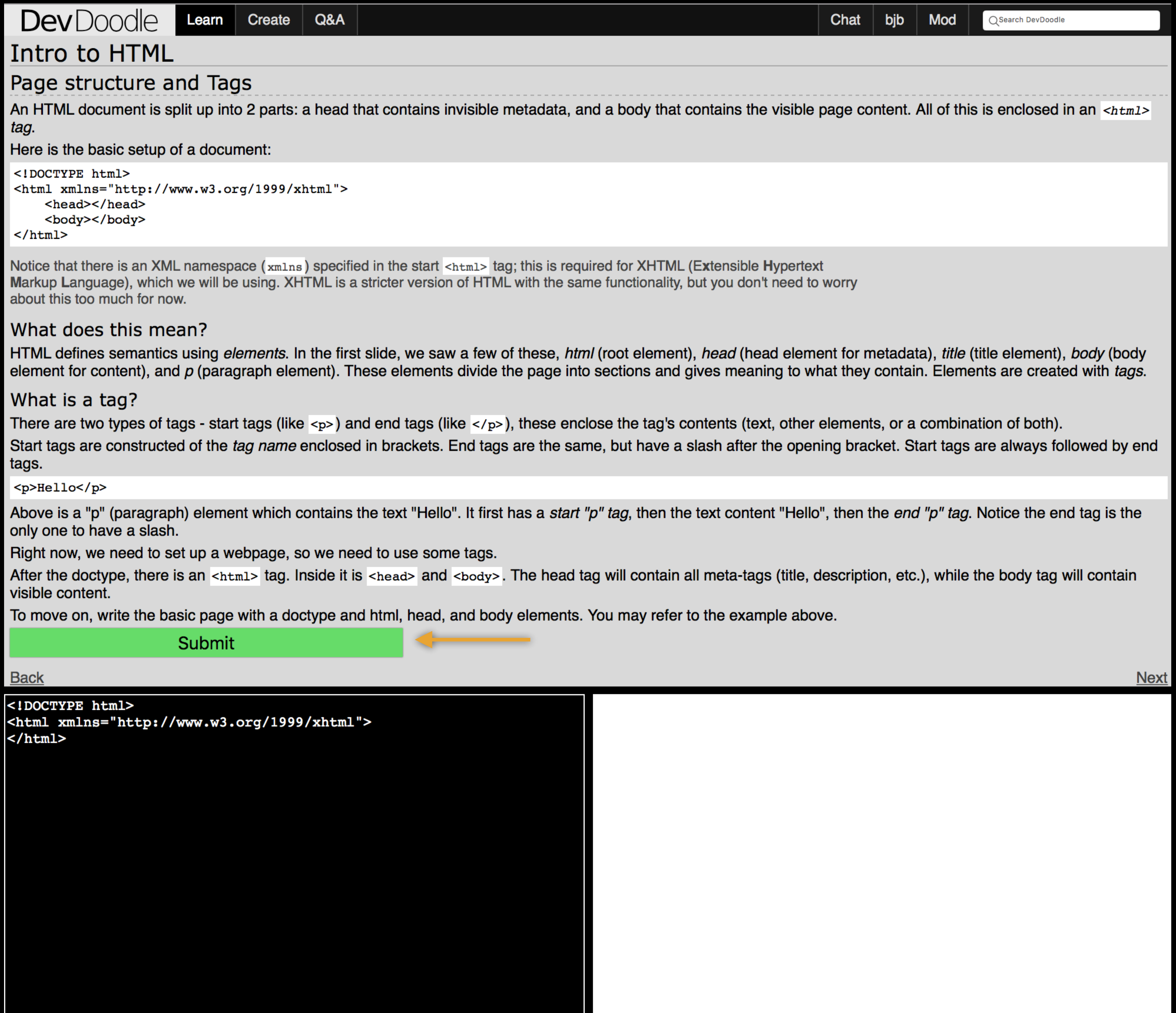
Back Next

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
</html>
```

DevDoodle

Sections

- Learn



Intro to HTML

Page structure and Tags

An HTML document is split up into 2 parts: a head that contains invisible metadata, and a body that contains the visible page content. All of this is enclosed in an `<html>` *tag*.

Here is the basic setup of a document:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
    <head></head>
    <body></body>
</html>
```

Notice that there is an XML namespace (`xmlns`) specified in the start `<html>` tag; this is required for XHTML (**E**x**t**ensible **H**ypertext **M**arkup **L**anguage), which we will be using. XHTML is a stricter version of HTML with the same functionality, but you don't need to worry about this too much for now.

What does this mean?

HTML defines semantics using *elements*. In the first slide, we saw a few of these, *html* (root element), *head* (head element for metadata), *title* (title element), *body* (body element for content), and *p* (paragraph element). These elements divide the page into sections and gives meaning to what they contain. Elements are created with *tags*.

What is a tag?

There are two types of tags - start tags (like `<p>`) and end tags (like `</p>`), these enclose the tag's contents (text, other elements, or a combination of both).

Start tags are constructed of the *tag name* enclosed in brackets. End tags are the same, but have a slash after the opening bracket. Start tags are always followed by end tags.

```
<p>Hello</p>
```

Above is a "p" (paragraph) element which contains the text "Hello". It first has a *start "p" tag*, then the text content "Hello", then the *end "p" tag*. Notice the end tag is the only one to have a slash.

Right now, we need to set up a webpage, so we need to use some tags.

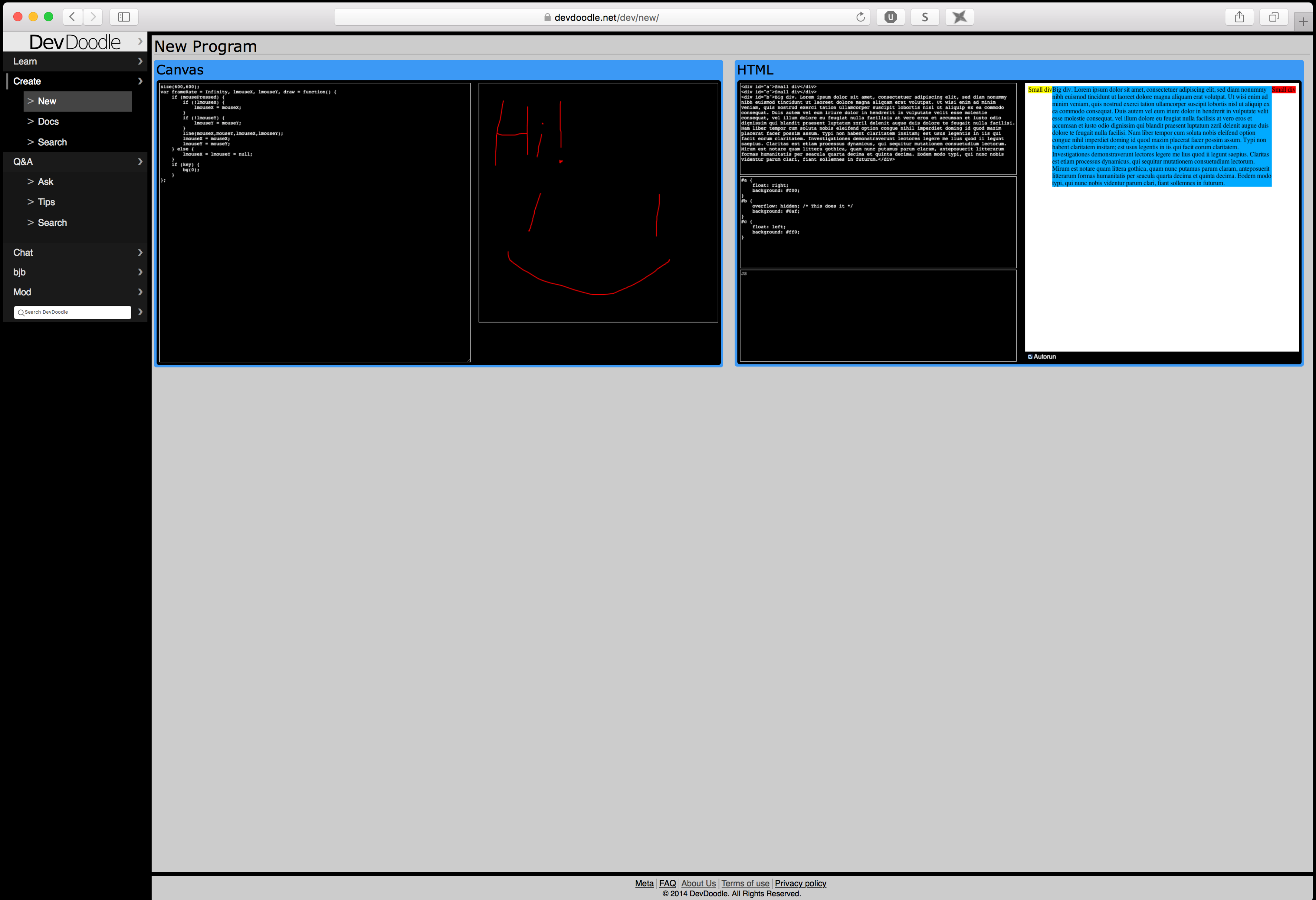
After the doctype, there is an `<html>` tag. Inside it is `<head>` and `<body>`. The head tag will contain all meta-tags (title, description, etc.), while the body tag will contain visible content.

To move on, write the basic page with a doctype and html, head, and body elements. You may refer to the example above.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
</html>
```

Sections

- Create



```
size(600,600);
var frameRate = Infinity, lmouseX, lmouseY, draw = function() {
    if (mousePressed) {
        if (!lmouseX) {
            lmouseX = mouseX;
        }
        if (!lmouseY) {
            lmouseY = mouseY;
        }
        line(mouseX,mouseY,lmouseX,lmouseY);
        lmouseX = mouseX;
        lmouseY = mouseY;
    } else {
        lmouseX = lmouseY = null;
    }
    if (key) {
        bg(0);
    }
};
```

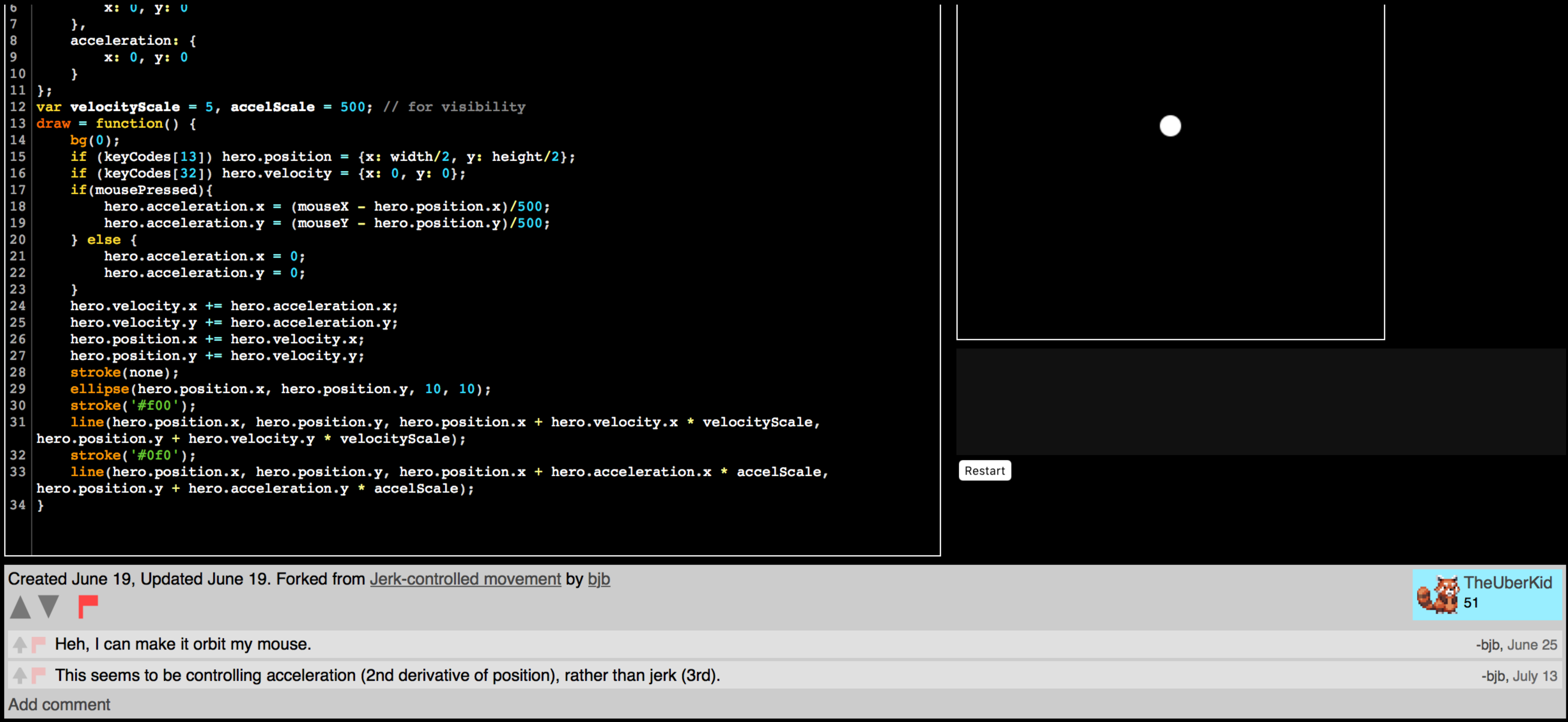
```
<div id="a">Small div</div>
<div id="c">Small div</div>
<div id="b">Big div. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy
nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim
veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo
consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie
consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio
dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.
Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim
placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui
facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt
saepius. Claritas est etiam processus dynamicus, qui sequitur mutationem consuetudium lectorum.
Mirum est notare quam littera gothica, quam nunc putamus parum claram, anteposuerit litterarum
formas humanitatis per seacula quarta decima et quinta decima. Eodem modo typi, qui nunc nobis
videntur parum clari, fiant sollemnes in futurum.</div>
```

```
#a {
    float: right;
    background: #f00;
}
#b {
    overflow: hidden; /* This does it */
    background: #0af;
}
#c {
    float: left;
    background: #ff0;
}
```

DevDoodle

Sections

- Create



```
            x: 0, y: 0
        },
        acceleration: {
            x: 0, y: 0
        }
};
var velocityScale = 5, accelScale = 500; // for visibility
draw = function() {
    bg(0);
    if (keyCodes[13]) hero.position = {x: width/2, y: height/2};
    if (keyCodes[32]) hero.velocity = {x: 0, y: 0};
    if(mousePressed){
        hero.acceleration.x = (mouseX - hero.position.x)/500;
        hero.acceleration.y = (mouseY - hero.position.y)/500;
    } else {
        hero.acceleration.x = 0;
        hero.acceleration.y = 0;
    }
    hero.velocity.x += hero.acceleration.x;
    hero.velocity.y += hero.acceleration.y;
    hero.position.x += hero.velocity.x;
    hero.position.y += hero.velocity.y;
    stroke(none);
    ellipse(hero.position.x, hero.position.y, 10, 10);
    stroke('#f00');
    line(hero.position.x, hero.position.y, hero.position.x + hero.velocity.x * velocityScale,
hero.position.y + hero.velocity.y * velocityScale);
    stroke('#0f0');
    line(hero.position.x, hero.position.y, hero.position.x + hero.acceleration.x * accelScale,
hero.position.y + hero.acceleration.y * accelScale);
}
```

Created June 19, Updated June 19. Forked from Jerk-controlled movement by bjb

TheUberKid
51

Heh, I can make it orbit my mouse. -bjb, June 25

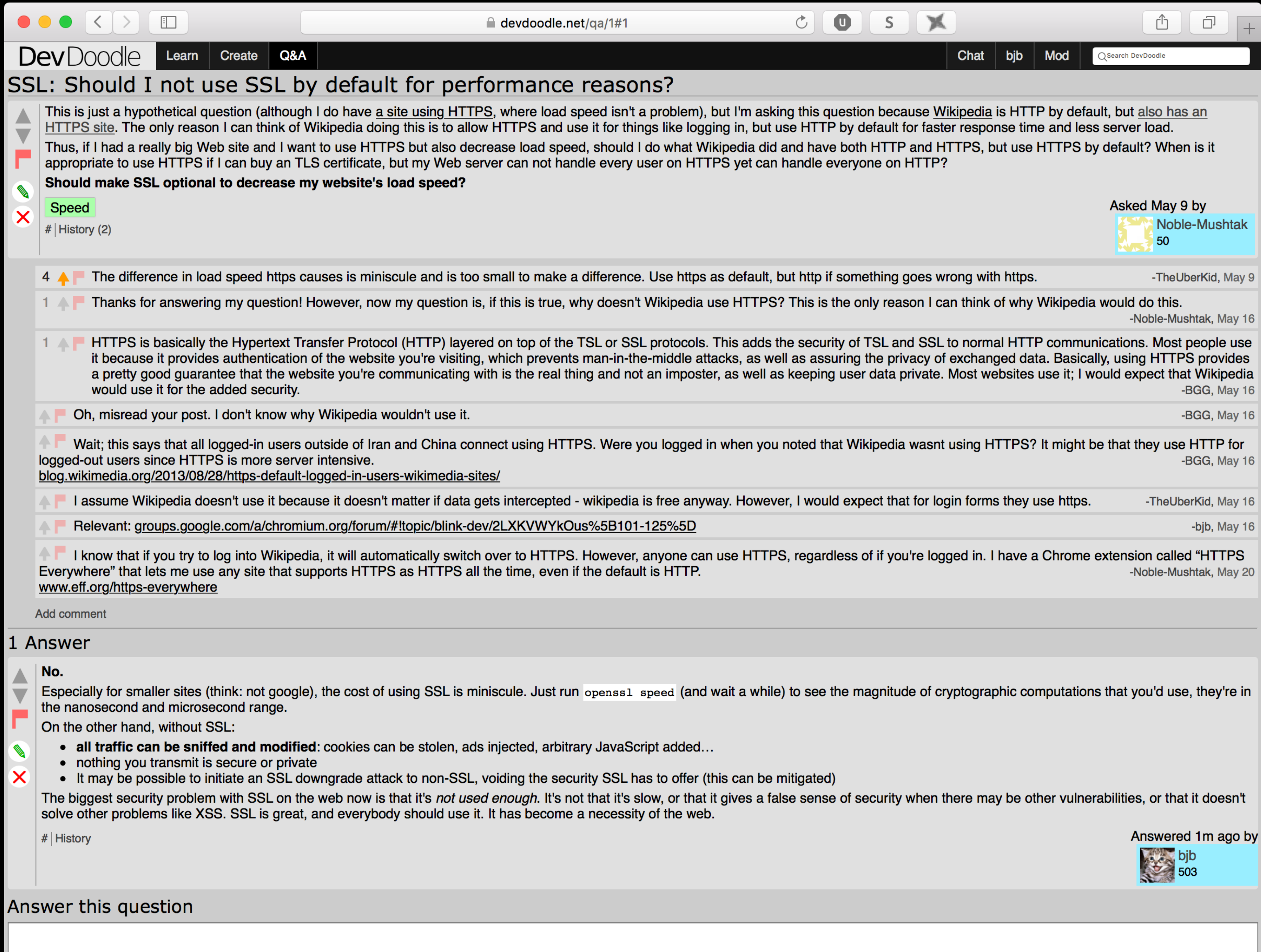
This seems to be controlling acceleration (2nd derivative of position), rather than jerk (3rd). -bjb, July 13

Add comment

DevDoodle

Sections

- Question & Answer



SSL: Should I not use SSL by default for performance reasons?

This is just a hypothetical question (although I do have a site using HTTPS, where load speed isn't a problem), but I'm asking this question because Wikipedia is HTTP by default, but also has an HTTPS site. The only reason I can think of Wikipedia doing this is to allow HTTPS and use it for things like logging in, but use HTTP by default for faster response time and less server load.

Thus, if I had a really big Web site and I want to use HTTPS but also decrease load speed, should I do what Wikipedia did and have both HTTP and HTTPS, but use HTTPS by default? When is it appropriate to use HTTPS if I can buy an TLS certificate, but my Web server can not handle every user on HTTPS yet can handle everyone on HTTP?

Should make SSL optional to decrease my website's load speed?

Speed

| History (2)

Asked May 9 by Noble-Mushtak 50

4 The difference in load speed https causes is miniscule and is too small to make a difference. Use https as default, but http if something goes wrong with https. -TheUberKid, May 9

1 Thanks for answering my question! However, now my question is, if this is true, why doesn't Wikipedia use HTTPS? This is the only reason I can think of why Wikipedia would do this. -Noble-Mushtak, May 16

1 HTTPS is basically the Hypertext Transfer Protocol (HTTP) layered on top of the TSL or SSL protocols. This adds the security of TSL and SSL to normal HTTP communications. Most people use it because it provides authentication of the website you're visiting, which prevents man-in-the-middle attacks, as well as assuring the privacy of exchanged data. Basically, using HTTPS provides a pretty good guarantee that the website you're communicating with is the real thing and not an imposter, as well as keeping user data private. Most websites use it; I would expect that Wikipedia would use it for the added security. -BGG, May 16

Oh, misread your post. I don't know why Wikipedia wouldn't use it. -BGG, May 16

Wait; this says that all logged-in users outside of Iran and China connect using HTTPS. Were you logged in when you noted that Wikipedia wasnt using HTTPS? It might be that they use HTTP for logged-out users since HTTPS is more server intensive. blog.wikimedia.org/2013/08/28/https-default-logged-in-users-wikimedia-sites/ -BGG, May 16

I assume Wikipedia doesn't use it because it doesn't matter if data gets intercepted - wikipedia is free anyway. However, I would expect that for login forms they use https. -TheUberKid, May 16

Relevant: groups.google.com/a/chromium.org/forum/#!topic/blink-dev/2LXKVWYkOus%5B101-125%5D -bjb, May 16

I know that if you try to log into Wikipedia, it will automatically switch over to HTTPS. However, anyone can use HTTPS, regardless of if you're logged in. I have a Chrome extension called “HTTPS Everywhere” that lets me use any site that supports HTTPS as HTTPS all the time, even if the default is HTTP. www.eff.org/https-everywhere -Noble-Mushtak, May 20

Add comment

1 Answer

No.

Especially for smaller sites (think: not google), the cost of using SSL is miniscule. Just run `openssl speed` (and wait a while) to see the magnitude of cryptographic computations that you'd use, they're in the nanosecond and microsecond range.

On the other hand, without SSL:

- **all traffic can be sniffed and modified**: cookies can be stolen, ads injected, arbitrary JavaScript added…
- nothing you transmit is secure or private
- It may be possible to initiate an SSL downgrade attack to non-SSL, voiding the security SSL has to offer (this can be mitigated)

The biggest security problem with SSL on the web now is that it's *not used enough*. It's not that it's slow, or that it gives a false sense of security when there may be other vulnerabilities, or that it doesn't solve other problems like XSS. SSL is great, and everybody should use it. It has become a necessity of the web.

| History

Answered 1m ago by bjb 503

Answer this question

DevDoodle

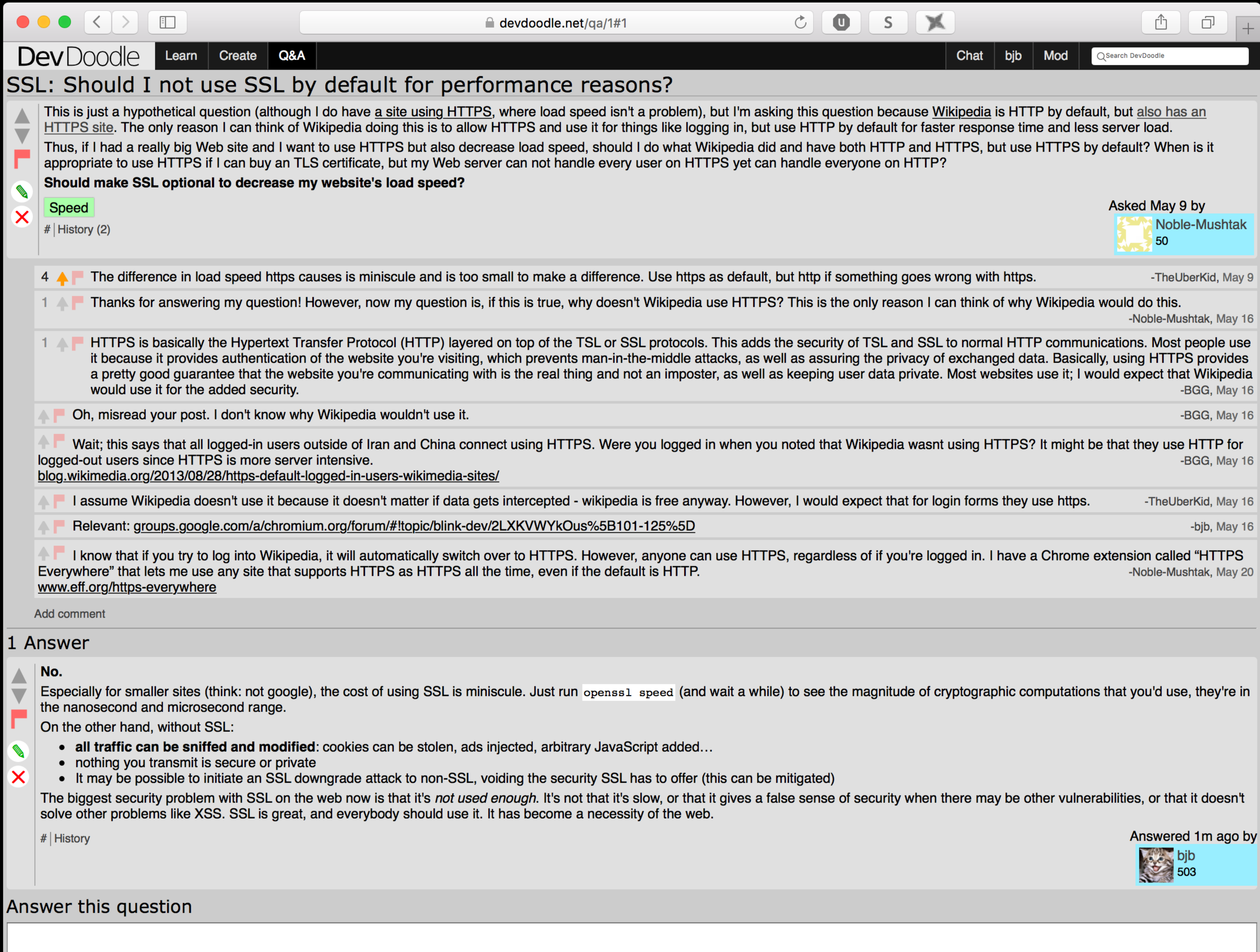
Sections

- Question & Answer

A focused *question*, not a request for help

Comments ask for clarification or give ideas

Potentially multiple answers, each share specific knowledge



SSL: Should I not use SSL by default for performance reasons?

This is just a hypothetical question (although I do have a site using HTTPS, where load speed isn't a problem), but I'm asking this question because Wikipedia is HTTP by default, but also has an HTTPS site. The only reason I can think of Wikipedia doing this is to allow HTTPS and use it for things like logging in, but use HTTP by default for faster response time and less server load.

Thus, if I had a really big Web site and I want to use HTTPS but also decrease load speed, should I do what Wikipedia did and have both HTTP and HTTPS, but use HTTPS by default? When is it appropriate to use HTTPS if I can buy an TLS certificate, but my Web server can not handle every user on HTTPS yet can handle everyone on HTTP?

Should make SSL optional to decrease my website's load speed?

Speed

| History (2)

Asked May 9 by Noble-Mushtak 50

4 The difference in load speed https causes is miniscule and is too small to make a difference. Use https as default, but http if something goes wrong with https. -TheUberKid, May 9

1 Thanks for answering my question! However, now my question is, if this is true, why doesn't Wikipedia use HTTPS? This is the only reason I can think of why Wikipedia would do this. -Noble-Mushtak, May 16

1 HTTPS is basically the Hypertext Transfer Protocol (HTTP) layered on top of the TSL or SSL protocols. This adds the security of TSL and SSL to normal HTTP communications. Most people use it because it provides authentication of the website you're visiting, which prevents man-in-the-middle attacks, as well as assuring the privacy of exchanged data. Basically, using HTTPS provides a pretty good guarantee that the website you're communicating with is the real thing and not an imposter, as well as keeping user data private. Most websites use it; I would expect that Wikipedia would use it for the added security. -BGG, May 16

Oh, misread your post. I don't know why Wikipedia wouldn't use it. -BGG, May 16

Wait; this says that all logged-in users outside of Iran and China connect using HTTPS. Were you logged in when you noted that Wikipedia wasnt using HTTPS? It might be that they use HTTP for logged-out users since HTTPS is more server intensive. -BGG, May 16
blog.wikimedia.org/2013/08/28/https-default-logged-in-users-wikimedia-sites/

I assume Wikipedia doesn't use it because it doesn't matter if data gets intercepted - wikipedia is free anyway. However, I would expect that for login forms they use https. -TheUberKid, May 16

Relevant: groups.google.com/a/chromium.org/forum/#!topic/blink-dev/2LXKVWYkOus%5B101-125%5D -bjb, May 16

I know that if you try to log into Wikipedia, it will automatically switch over to HTTPS. However, anyone can use HTTPS, regardless of if you're logged in. I have a Chrome extension called "HTTPS Everywhere" that lets me use any site that supports HTTPS as HTTPS all the time, even if the default is HTTP. -Noble-Mushtak, May 20
www.eff.org/https-everywhere

Add comment

1 Answer

No.

Especially for smaller sites (think: not google), the cost of using SSL is miniscule. Just run `openssl speed` (and wait a while) to see the magnitude of cryptographic computations that you'd use, they're in the nanosecond and microsecond range.

On the other hand, without SSL:

- **all traffic can be sniffed and modified**: cookies can be stolen, ads injected, arbitrary JavaScript added…
- nothing you transmit is secure or private
- It may be possible to initiate an SSL downgrade attack to non-SSL, voiding the security SSL has to offer (this can be mitigated)

The biggest security problem with SSL on the web now is that it's *not used enough*. It's not that it's slow, or that it gives a false sense of security when there may be other vulnerabilities, or that it doesn't solve other problems like XSS. SSL is great, and everybody should use it. It has become a necessity of the web.

| History

Answered 1m ago by bjb 503

Answer this question

DevDoodle

Sections

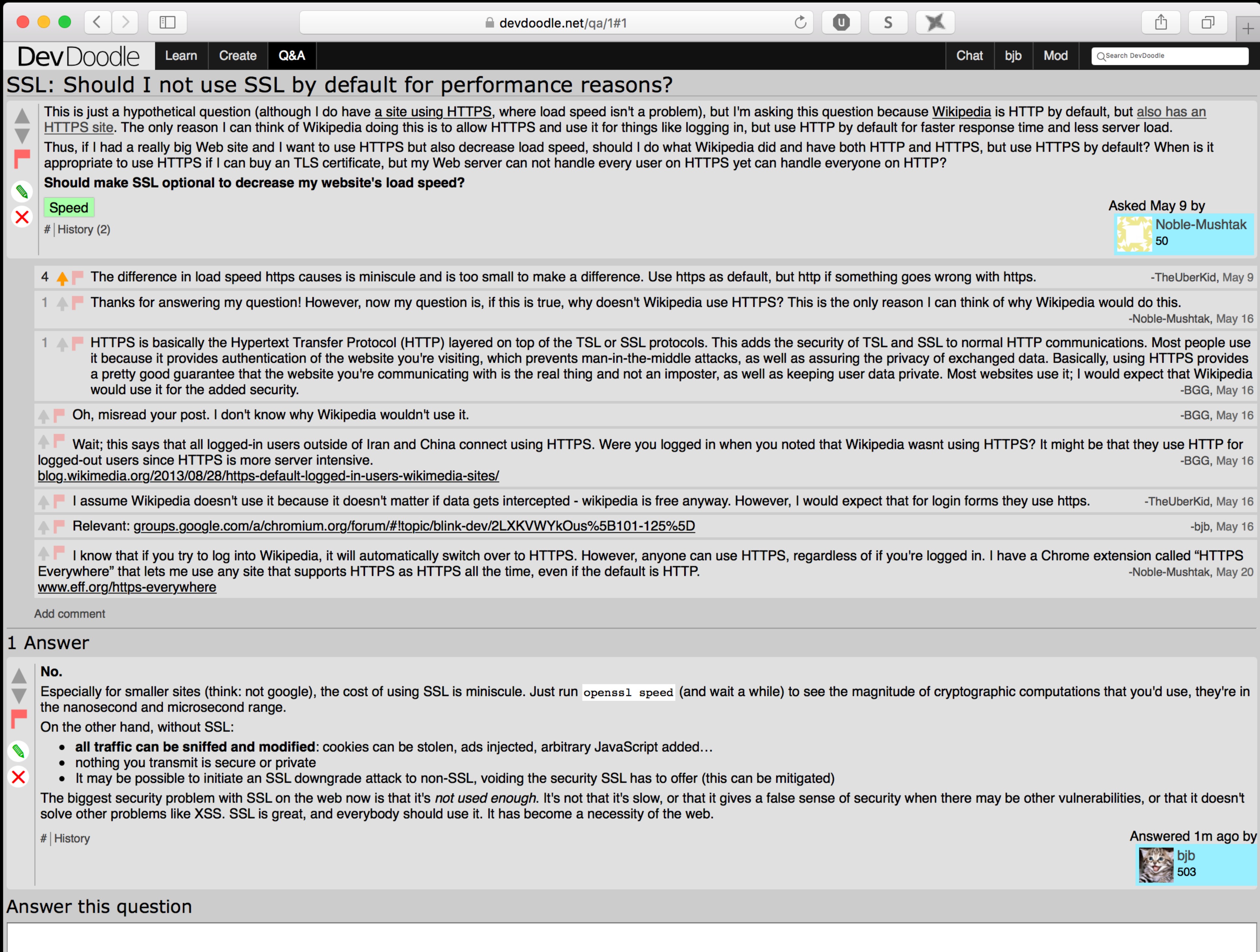
- Question & Answer

A focused *question*, not a request for help

Comments ask for clarification or give ideas

Potentially multiple answers, each share specific knowledge

(google juice!)


devdoodle.net/qa/1#1

DevDoodle | Learn | Create | Q&A | Chat | bjb | Mod | Search DevDoodle

SSL: Should I not use SSL by default for performance reasons?

This is just a hypothetical question (although I do have a site using HTTPS, where load speed isn't a problem), but I'm asking this question because Wikipedia is HTTP by default, but also has an HTTPS site. The only reason I can think of Wikipedia doing this is to allow HTTPS and use it for things like logging in, but use HTTP by default for faster response time and less server load.

Thus, if I had a really big Web site and I want to use HTTPS but also decrease load speed, should I do what Wikipedia did and have both HTTP and HTTPS, but use HTTPS by default? When is it appropriate to use HTTPS if I can buy an TLS certificate, but my Web server can not handle every user on HTTPS yet can handle everyone on HTTP?

Should make SSL optional to decrease my website's load speed?

Speed

| History (2)

Asked May 9 by Noble-Mushtak 50

4 The difference in load speed https causes is miniscule and is too small to make a difference. Use https as default, but http if something goes wrong with https. -TheUberKid, May 9

1 Thanks for answering my question! However, now my question is, if this is true, why doesn't Wikipedia use HTTPS? This is the only reason I can think of why Wikipedia would do this. -Noble-Mushtak, May 16

1 HTTPS is basically the Hypertext Transfer Protocol (HTTP) layered on top of the TSL or SSL protocols. This adds the security of TSL and SSL to normal HTTP communications. Most people use it because it provides authentication of the website you're visiting, which prevents man-in-the-middle attacks, as well as assuring the privacy of exchanged data. Basically, using HTTPS provides a pretty good guarantee that the website you're communicating with is the real thing and not an imposter, as well as keeping user data private. Most websites use it; I would expect that Wikipedia would use it for the added security. -BGG, May 16

Oh, misread your post. I don't know why Wikipedia wouldn't use it. -BGG, May 16

Wait; this says that all logged-in users outside of Iran and China connect using HTTPS. Were you logged in when you noted that Wikipedia wasnt using HTTPS? It might be that they use HTTP for logged-out users since HTTPS is more server intensive. blog.wikimedia.org/2013/08/28/https-default-logged-in-users-wikimedia-sites/ -BGG, May 16

I assume Wikipedia doesn't use it because it doesn't matter if data gets intercepted - wikipedia is free anyway. However, I would expect that for login forms they use https. -TheUberKid, May 16

Relevant: groups.google.com/a/chromium.org/forum/#!topic/blink-dev/2LXKVWYkOus%5B101-125%5D -bjb, May 16

I know that if you try to log into Wikipedia, it will automatically switch over to HTTPS. However, anyone can use HTTPS, regardless of if you're logged in. I have a Chrome extension called "HTTPS Everywhere" that lets me use any site that supports HTTPS as HTTPS all the time, even if the default is HTTP. www.eff.org/https-everywhere -Noble-Mushtak, May 20

Add comment

1 Answer

No.

Especially for smaller sites (think: not google), the cost of using SSL is miniscule. Just run `openssl speed` (and wait a while) to see the magnitude of cryptographic computations that you'd use, they're in the nanosecond and microsecond range.

On the other hand, without SSL:

- **all traffic can be sniffed and modified**: cookies can be stolen, ads injected, arbitrary JavaScript added…
- nothing you transmit is secure or private
- It may be possible to initiate an SSL downgrade attack to non-SSL, voiding the security SSL has to offer (this can be mitigated)

The biggest security problem with SSL on the web now is that it's *not used enough*. It's not that it's slow, or that it gives a false sense of security when there may be other vulnerabilities, or that it doesn't solve other problems like XSS. SSL is great, and everybody should use it. It has become a necessity of the web.

| History

Answered 1m ago by bjb 503

Answer this question

DevDoodle

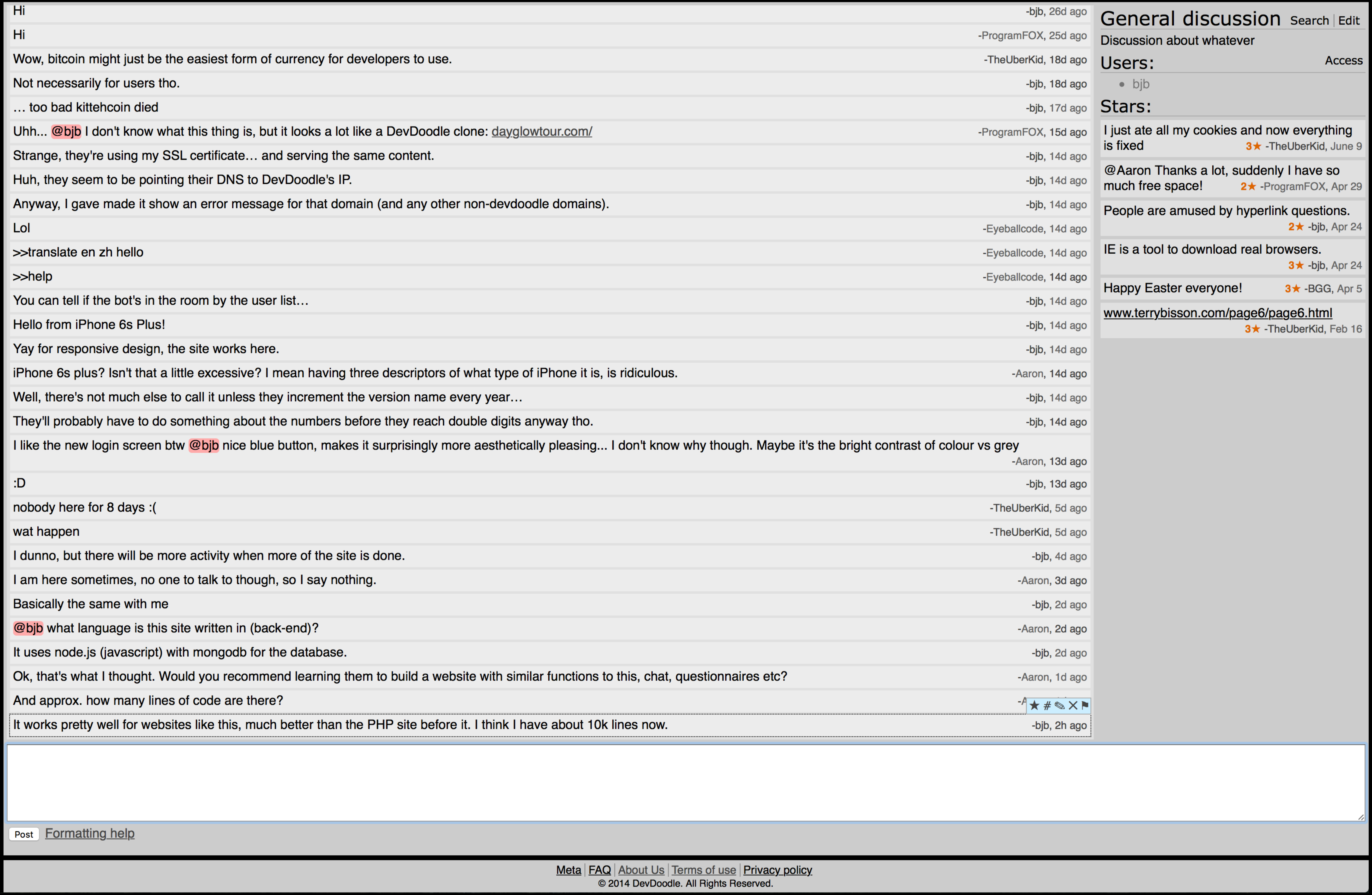
Sections

- Question & Answer
  - General Reference Question (long, broad)
  - Specific questions (still not debugging)
    - MCVE
  - Linkage

DevDoodle

Sections

- Chat



Hi -bjb, 26d ago
Hi -ProgramFOX, 25d ago
Wow, bitcoin might just be the easiest form of currency for developers to use. -TheUberKid, 18d ago
Not necessarily for users tho. -bjb, 18d ago
… too bad kittehcoin died -bjb, 17d ago
Uhh... @bjb I don't know what this thing is, but it looks a lot like a DevDoodle clone: dayglowtour.com/ -ProgramFOX, 15d ago
Strange, they're using my SSL certificate… and serving the same content. -bjb, 14d ago
Huh, they seem to be pointing their DNS to DevDoodle's IP. -bjb, 14d ago
Anyway, I gave made it show an error message for that domain (and any other non-devdoodle domains). -bjb, 14d ago
Lol -Eyeballcode, 14d ago
>>translate en zh hello -Eyeballcode, 14d ago
>>help -Eyeballcode, 14d ago
You can tell if the bot's in the room by the user list… -bjb, 14d ago
Hello from iPhone 6s Plus! -bjb, 14d ago
Yay for responsive design, the site works here. -bjb, 14d ago
iPhone 6s plus? Isn't that a little excessive? I mean having three descriptors of what type of iPhone it is, is ridiculous. -Aaron, 14d ago
Well, there's not much else to call it unless they increment the version name every year… -bjb, 14d ago
They'll probably have to do something about the numbers before they reach double digits anyway tho. -bjb, 14d ago
I like the new login screen btw @bjb nice blue button, makes it surprisingly more aesthetically pleasing... I don't know why though. Maybe it's the bright contrast of colour vs grey -Aaron, 13d ago
:D -bjb, 13d ago
nobody here for 8 days :(-TheUberKid, 5d ago
wat happen -TheUberKid, 5d ago
I dunno, but there will be more activity when more of the site is done. -bjb, 4d ago
I am here sometimes, no one to talk to though, so I say nothing. -Aaron, 3d ago
Basically the same with me -bjb, 2d ago
@bjb what language is this site written in (back-end)? -Aaron, 2d ago
It uses node.js (javascript) with mongodb for the database. -bjb, 2d ago
Ok, that's what I thought. Would you recommend learning them to build a website with similar functions to this, chat, questionnaires etc? -Aaron, 1d ago
And approx. how many lines of code are there? -A
It works pretty well for websites like this, much better than the PHP site before it. I think I have about 10k lines now. -bjb, 2h ago

General discussion Search | Edit
Discussion about whatever
Users: Access
- bjb

Stars:
I just ate all my cookies and now everything is fixed 3★ -TheUberKid, June 9
@Aaron Thanks a lot, suddenly I have so much free space! 2★ -ProgramFOX, Apr 29
People are amused by hyperlink questions. 2★ -bjb, Apr 24
IE is a tool to download real browsers. 3★ -bjb, Apr 24
Happy Easter everyone! 3★ -BGG, Apr 5
www.terrybisson.com/page6/page6.html 3★ -TheUberKid, Feb 16

Meta | FAQ | About Us | Terms of use | Privacy policy
© 2014 DevDoodle. All Rights Reserved.

Summary

- Community culture and processes
- Dichotomy of OP vs search engine user
- Attention market; needs to incentivize
- Self-sufficient; privilege system

DevDoodle

A developer network